ЗАДАЧА ВЫБОРА ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ, ОБЕСПЕЧИВАЮЩИХ ЗАЩИТУ ИНФОРМАЦИИ, ДЛЯ СЕРВЕРОВ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ И АЛГОРИТМЫ ЕЕ РЕШЕНИЯ

Быков А.Ю.[1], Крыгин И.А.[2], Гришунин М.В.[3]

Цель статьи: *обеспечение информационной безопасности на серверах различного назначения автоматизированной системы на основе формулировки оптимизационной постановки задачи выбора вспомогательных процессов для защиты информации, разработки и исследования алгоритмов решения этой задачи.*

Метод: *для решения задачи предложены два точных алгоритма неполного перебора с экспоненциальной вычислительной сложностью, основанных на идеях метода Балаша. Один начинает перебор с решения, состоящего из всех единиц, второй с решения, состоящего из всех нулей. Также предложены два приближенных алгоритма с полиномиальной сложностью, основанных на идеях «жадного» алгоритма, один алгоритм начинает поиск с нулевого решения, другой, с единичного решения.*

Полученный результат: *получена математическая модель и алгоритмы решения задачи выбора процессов для защиты информации в условиях ограниченных вычислительных ресурсов серверов. Модель выбора процессов является задачей булевого программирования с нелинейным показателем качества и линейными ограничениями. Показатель задает оценку предотвращенного ущерба при использовании выбранных процессов с учетом вероятности или возможности проведения различных атак на серверы, ценности, хранимые данные, и вероятности защиты от атак с помощью процессов. В ходе экспериментов с целью уменьшения времени решения задачи разработаны рекомендации по выбору одного из двух алгоритмов (поиск, начиная с нулевого решения, и поиск, начиная с единичного решения) среди пар точных и приближенных алгоритмов в зависимости от обеспеченности ресурсами.*

Ключевые слова: *информационная безопасность, дискретная оптимизация, булево программирование, коэффициент обеспеченности ресурсами, вычислительная сложность алгоритма, приближенное решение.*

DOI:10.21681/2311-3456-2020-05-30-44

Введение

Оптимизационные модели часто используются для выбора в автоматизированных системах средств защиты (СЗ) информации как программных или аппаратных, так организационных и других. Рассмотрим некоторые примеры подобных моделей применительно к задачам защиты информации.

В [1] предложен выбор СЗ по разным показателям с оценкой относительной значимости каждого показателя. В качестве показателей используются: стоимость СЗ, надежность СЗ, удобство пользовательского интерфейса, быстродействие системы после внедрения СЗ. В [2] подобный подход используется для выбора средств криптографической защиты для систем дистанционного банковского обслуживания по технологии «толстый клиент». В [3] для многокритериального выбора СЗ от несанкционированного доступа используется метод анализа иерархий.

В [4] формулируется задача оптимизации выбора СЗ с применением Марковской модели угроз, для ее решения предложен метод последовательного анализа вариантов. Используется показатель, названный средним временем жизни защищаемой системы. Получена аналитическая формула для среднего времени жизни системы, выраженная через вероятности реализации угроз и вероятности их предотвращения СЗ.

В [5] решалась задача выбора антивирусных программ разных производителей для узлов вычислительной сети. Предложено согласовывать как частные решения различных антивирусных программ по обнаружению или отсутствию вредоносного кода, так и согласованные решения, позволяющие снизить ошибки обнаружения первого и второго родов.

В [6] задача выбора варианта СЗ математически формализована в виде многокритериальной задачи оп-

1 Быков Александр Юрьевич, кандидат технических наук, доцент, доцент кафедры «Информационная безопасность», МГТУ им. Н.Э. Баумана, г. Москва, Россия. E-mail: abykov@bmstu.ru

2 Крыгин Иван Александрович, аспирант кафедры «Информационная безопасность», МГТУ им. Н.Э. Баумана, г. Москва, Россия. E-mail: krygin.ia@gmail.com

3 Гришунин Максим Вадимович, аспирант кафедры «Информационная безопасность», МГТУ им. Н.Э. Баумана, г. Москва, Россия. E-mail: grishunin-mv@ya.ru

тимизации, для решения которой разработан алгоритм на основе метода вектора спада.

В [7] оптимизация защиты выполняется на основе построения дерева атак, в котором цели и (или) подцели нарушителя группируются с использованием логические операций «И» и «ИЛИ». Оптимизация защиты основана на выявлении по дереву наиболее критичных атак, которые должны быть предотвращены.

В [8] выполняется многокритериальная оптимизация при приеме-передачи в сетях с помехами с множеством входов и множеством выходов. Используются следующие показатели и критерии: качество передачи сигнала, минимизация энергии и повышение безопасности. Для решения многокритериальной задачи предложена свертка Чебышева, также выявляются оптимальные по Парето решения. Для решения однокритериальных задач используются функции штрафа и решаются выпуклые задачи оптимизации.

В [9] проводится оптимизация нейросети для системы обнаружения вторжений. Используется глубокая сеть доверия (deep belief network), для оптимизации структуры сети совместно используются методы: роя частиц (particle swarm optimization), метод, основанный на поведении косяка рыб (the fish swarm behavior of cluster, foraging), а также генетические алгоритмы.

В [10] также исследуются системы обнаружения вторжений, для повышения точности классификации искусственной нейронной сети, совестно используются метод магнитной оптимизации (Magnetic Swarm Optimization) и метод роя частиц.

В [11] для выявления факта использования стеганографии в видео данных представлен подход на основе векторов движения, который не зависит от детального знания алгоритмов встраивания. В большинстве современных стандартов кодирования видео каждый вектор движения является локально-оптимальным в смысле искажения скорости, и любая модификация неизбежно сместит вектор движения с локально-оптимального на неоптимальный, это используется для выявления факта стеганографического встраивания.

В [12] проводится оптимизация схемы формирования диаграммы направленности для одновременной безопасной передачи данных на несколько приемников информации. Предложено два критерия: минимизация общей мощности передачи и гарантия надежности передачи данных. Задача оптимизации является невыпуклой из-за случайных ограничений, предложены два преобразования исходной задачи, основанные на методах безопасной выпуклой аппроксимации (safe-convex-approximation techniques).

В [13] оптимизируется траектория и мощности передачи базовых станций и ретрансляторов в системе беспилотных летательных аппаратов. В качестве критерия используется максимизация уровня секретности по всем приемникам информации. Задача оптимизации невыпуклая, а переменные оптимизации связаны, что приводит к тому, что задача оптимизации математически неразрешима. Предлагается декомпозиция задачи оптимизации на две подзадачи и решение их, используя итерационный алгоритм и метод последовательного выпуклого приближения (the successive convex approximation technique).

В [14] решается задача безопасной передачи в беспроводной сети с ретрансляцией, где пара законных пользователей обменивается данными с помощью сети с множеством входов и выходов, в сети присутствуют перехватчики сообщений. В качестве показателей используются: мощность источника, а также отношение сигнал/шум. Для решения результирующей невыпуклой задачи оптимизации с заданной сложностью предложен новый разностный-выпуклый (difference-of-convex) алгоритм на основе штрафных функций.

В [15-17] рассмотрены игровые постановки задачи и некоторые алгоритмы их решения применительно к оптимизационному выбору объектов для защиты и моделированию выбора объектов для атаки стороны нападения. Рассмотрены примеры дискретной и непрерывной задач, каждый игрок для оптимизации должен решать свою задачу линейного или дискретного программирования.

В [18] для распознавания ботов в социальных сетях используется алгоритм «случайный лес». В [19] рассмотрены детерминированные методы построения графов Рамануджана, предназначенные для применения в криптографических алгоритмах, основанных на обобщённых клеточных автоматах.

Ниже представлена постановка задачи оптимизационного выбора процессов (программ или приложений) для защиты серверов распределенной вычислительной сети при ограничениях на ресурсы и некоторые алгоритмы ее решения.

1. Содержательная постановка задачи

Рассмотрим распределенную вычислительную систему, состоящую из множества серверов, предоставляющих различные сервисы в сети. Сервера решают некоторые целевые задачи, для которых они предназначены. На серверах установлено необходимое программное обеспечение, включая операционную систему (ОС) и прикладные программы. Непосредственные услуги потребителям оказывают прикладные программы (для названия работающей программы будет использовать термин вычислительный процесс), постоянно выполняющиеся на серверах. Существует минимальная конфигурация прикладных и системных программ, обеспечивающих выполнение каждым сервером своих целевых задач, эти программы выполняются в виде процессов.

Кроме основных процессов (без запуска, которых не смогут решаться целевые задачи) запускаются обеспечивающие процессы, например, процессы для обеспечения безопасности информации, такие как антивирусные программы, средства для предотвращения DDoS, программы контроля целостности данных, программы для обеспечения конфиденциальности, защиты от несанкционированных сетевых подключений, системы обнаружения вторжений и т.д. Подобные программы могут запускаться как отдельные процессы или, например, как службы операционной системы Windows (некоторые службы, входящие в состав Windows могут запускаться или быть отключены).

DOI:10.21681/2311-3456-2020-05-30-44

Минимальная конфигурация прикладных процессов для целевых задач, а также необязательные служебные процессы требует наличия определенных вычислительных ресурсов на сервере (времени и памяти), это ресурсы назовем частными ресурсами каждого сервера. Каждый запущенный процесс загружает центральный процессор (или процессоры для многопроцессорной системы), загрузку можно измерять в процентах загрузки процессора (как в диспетчере задач), эта загрузка зависит от быстродействия текущего сервера, также процесс расходует оперативную память. Минимальная конфигурация, как правило, использует не 100 % загрузки процессора и памяти. Служебные или обеспечивающие процессы также требуют процессорного времени и памяти, и все не могут быть одновременно запущены из-за нехватки ресурсов, как правило (в этом и нет смысла), возникает задача выбора используемых программ (процессов).

Кроме частных ресурсов каждого сервера будем рассматривать общие ресурсы для всех серверов, например, это может быть финансовый ресурс, выделенный для закупки служебных программ.

На информационные ресурсы серверов системы могут проводиться атаки как способы реализации угроз с различными целями. Можно выделять различные виды атак, например, DDoS атаки, атаки с целью доступа к конфиденциальной информации, атаки для искажения данные (нарушение целостности) и т.д. Существуют оценки возможного ущерба для владельца серверов в случае успешности этих атак для каждого сервера с учетом важности хранимой или обрабатываемой информации на сервере. Использование служебных процессов позволяет защитить сервера от проводимых атак полностью или частично или уменьшить последствия атак (возможный ущерб).

Обобщенно сформулируем содержательную постановку задачи выбора служебных процессов (программ) применительно к защите информации.

Дано:

- Множество серверов, решающих целевые задачи.
- Множество служебных вычислительных процессов для обеспечения информационной безопасности (могут быть заданы независимыми приложениями или службами, входящими в состав ОС).
- Множество возможных типов атак, проводимых на информационные ресурсы системы.
- Для каждого сервера и каждого типа атаки задана оценка ущерба для владельца сервера в случае успешной атаки с учетом важности информации, хранимой или обрабатываемой на сервере.
- Для каждого служебного процесса и типа атаки заданы вероятность или возможности предотвращения (в терминах нечетких множеств) атаки при работе процесса или возможная степень уменьшения ущерба при использовании служебного процесса.
- Для каждого сервера и каждого служебного процесса заданы вычислительные ресурсы, требуемые для запуска процесса на сервере:
  - процент загрузки процессора (процессоров) на сервере, который зависит от вычислительной мощности сервера и параметров процесса;
  - расход оперативной памяти.

Задана «стоимость процесса», если требуется покупка соответствующей программы.

Заданы ресурсы каждого сервера – размер памяти и вычислительная производительность сервера, которые могут быть выделены на служебные процессы с учетом того, что часть ресурсов использованы основными прикладными процессами, а также денежные средства на покупку служебных программ (при нормированном подходе производительность измеряется в %, вся производительность сервера – 100 %).

Найти:

С целью минимизации возможного ущерба определить, какие служебные процессы должны быть запущены на каждом сервере, чтобы выполнялись ограничения на вычислительную производительность серверов, оперативную память и финансовые ресурсы стороны защиты на покупку служебного программного обеспечения.

Рассмотрим математическую постановку задачи.

2. Математическая постановка задачи

Базисные множества

1. $S=\{s_1, s_2, \ldots, s_n\}$ – множество объектов (серверов) в распределённой системе, $N=\{1,2,\ldots,n\}$ – множество индексов этих объектов.
2. $A=\{a_1,a_2,\ldots,a_m\}$ – множество типов атак (реализаций угроз безопасности) в защищаемой системе, $M=\{1,2,\ldots,m\}$ – множество индексов этих атак.
3. $\Pi=\{\pi_1, \pi_2,\ldots,\pi_\lambda\}$ – множество служебных процессов (программ), которые могут быть использованы на объектах (запущены на серверах) для обеспечения безопасности, $\Lambda=\{1,2,\ldots,\lambda\}$ – множество индексов этих процессов.
4. $R=\{r_1,r_1,\ldots,r_l\}$ – множество ограниченных ресурсов, выделенных на служебные процессы, $L=\{1,2,\ldots,l\}$ – множество индексов этих ресурсов.

Множество ресурсов R делится на два непересекающихся подмножества $R = R^{(ч)} \cup R^{(о)}$, $R^{(ч)}$ – множество частных ресурсов для серверов (у каждого сервера свои ресурсы), $R^{(о)}$ – множество общих ресурсов для серверов, такой ресурс распределяется между всеми серверами. Примерами частных ресурсов для серверов являются ресурсы производительности и памяти для каждого сервера. Примером общего ресурса является финансовый ресурс. Для частных и общих ресурсов введем отдельную нумерацию: $L^{(ч)} = \{1, 2, \ldots, l^{(ч)}\}$, $L^{(о)} = \{1, 2, \ldots, l^{(о)}\}$ – множества индексов частных и общих ресурсов, соответственно.

Параметры элементов множеств и отношений между ними

1. $w_{ij} \geq 0, \forall\, i \in N,\ j \in M$ – оценка ущерба защитника на *i*-ом сервере при реализации *j*-го типа атаки.
2. $p_{ij}^{(a)}\ \forall\ i \in N,\ j \in M$ – вероятность (или возможность) реализации *j*-ой атаки на *i*-ый сервер за заданный интервал времени.
3. $p_{jk} \in [0,1], \forall\, j \in M,\ k \in \Lambda$ – вероятность (или возможность) предотвращения *j*-ой атаки (реали-

зации угрозы) при использовании *k*-го процесса на любом из серверов или степень снижения ущерба защитника.

4. $v_{ijk}^{(ч)} \geq 0,\ \forall i \in N,\ j \in L^{(ч)},\ k \in \Lambda$ – значение *j*-го частного ресурса (производительности и памяти) требуемого для работы *k*-го процесса *i*-ом сервере.
5. $V_{ij}^{(ч)} \geq 0, \forall i \in N,\ j \in L^{(ч)}$ – максимальное значение *j*-го частного ресурса на *i*-ом сервере, которое может быть использовано для служебных процессов, с учетом расхода этого ресурса на функционирование основных процессов, обеспечивающих решение целевых задач.
6. $v_{ijk}^{(о)} \geq 0,\ \forall i \in N,\ j \in L^{(о)},\ k \in \Lambda$ – значение j-го общего ресурса (например, финансового) требуемого для работы *k*-го процесса *i*-ом сервере.
7. $V_j^{(о)} \geq 0, \forall\ j \in L^{(о)}$ – максимальное значение *j*-го общего ресурса для всех серверов, которое может быть использовано для служебных процессов.
8. Некоторые вычислительные процессы (приложения) объединим в однотипные группы, из группы имеет смысл устанавливать только одно приложение, в каждой группе более одного приложения. Например, может существовать несколько антивирусных приложений, из них имеет смысл установить только одно, или несколько средств криптографической защиты и т.д. $G = \{1, 2, \dots, g\}$ – индексы этих групп. Для определения таких групп введем булевую матрицу $B_{<g\ x\ \lambda>} \|b_{ik}\|, i \in G, k \in \Lambda, b_{ik} = 1,$ если *k*-ое приложение входит в *i*-ую группу, $b_{ik} = 0$ – в противном случае.

Искомые переменные

Введем булеву переменную $x_{ik} \in \{0,\ 1\}, \forall\ i \in N,\ k \in \Lambda,\ x_{ik} = 1$, если *k*-ый процесс запущен *i*-ом сервере, $x_{ik} = 0$ – в противном случае, переменные образуют вектор X (так как каждая компонента имеет два индекса, то логичнее компоненты x_{ik} записать в виде матрицы, но в описании алгоритмов будем для удобства использовать сплошную индексацию одним индексом, в этом случае эту матрицу можно записать в виде вектора).

Показатель качества

В качестве показателя качества выбора будем использовать оценку возможного предотвращенного ущерба в течение заданного интервала времени. При расчете ущерба будем полагать, что вероятность (степень защищенности при нечетком описании) защиты *i*-го сервера от *j*-го типа атаки при использовании нескольких служебных процессов определяется процессом с максимальной вероятностью: $P_{ij}(X) = \max\limits_{k \in \Lambda}\{p_{jk} x_{ik}\},\quad \forall\ i \in N,\ j \in M$. Тогда оценка предотвращенного ущерба для всех серверов:

$$U(X) = \sum_{i \in N} \sum_{j \in M} w_{ij}\ P_{ij}(X)\ p_{ij}^{(а)} = \\ = \sum_{i \in N} \sum_{j \in M} w_{ij}\ \max\limits_{k \in \Lambda}\{p_{jk} x_{ik}\}\ p_{ij}^{(а)}. \quad (1)$$

Данный показатель необходимо максимизировать.

Ограничения

Ограничения на использование частных ресурсов на каждом сервере:

$$\sum_{k \in \Lambda} v_{ijk}^{(ч)}\ x_{ik} \leq V_{ij}^{(ч)}, \forall\ i \in N,\ j \in L^{(ч)}. \quad (2)$$

Ограничение на использование общих ресурсов для всех серверов:

$$\sum_{i \in N} \sum_{k \in \Lambda} v_{ijk}^{(о)}\ x_{ik} \leq V_j^{(о)},\ \forall\ j \in L^{(о)}. \quad (3)$$

Ограничение на то, что на любом сервере нельзя устанавливать более одного приложения из каждой группы:

$$\sum_{k \in \Lambda} b_{jk}\ x_{ik} \leq 1,\ \forall\ i \in N,\ j \in G. \quad (4)$$

Представленная постановка задачи с показателем (1) и ограничениями (2)-(4) является задачей булевого программирования с нелинейным по X показателем (1) и линейными ограничениями (2)-(4).

3. Алгоритмы решения задачи

3.1. Точные алгоритмы

Рассмотрим два точных метода, основанные на неполном (частичном) переборе решений на дереве. Будем учитывать две следующих аксиомы, основанные на особенностях поставленной задачи.

1. Если некоторый вектор X, содержащий 0 и 1, по ограничениям (2) или (3) недопустим, то в случае замены любого 0 на 1, полученный вектор будет также недопустимым.
2. Если в некотором векторе X, содержащем 0 и 1 (неважно допустимый это вектор по ограничениям или нет), любую 1 заменить на 0, то значение показателя качества (1) не увеличится.

Первый алгоритм основан на частичном переборе, начиная с единичного решения (X состоит из всех 1), считаем, что это решение по ограничениям (2)-(4) недопустимое, в противном случае, она будет оптимальным, и задача становится тривиальной. Второй алгоритм, основан на частичном переборе, начиная с нулевого решения (X состоит из всех 0), решение по ограничением (2)-(4) допустимое, но не оптимальное, алгоритм базируется на идеях метода Балаша [20].

Алгоритм частичного перебора, начиная с единичного решения

Первоначально считаем, что начальное решение состоит из всех 1 и является недопустимым по ограничениям, рекордное значение показателя качества 0, рекордное решение пустое. На каждом шаге алгоритма для недопустимого решения X получаем новые решения, заменяя последовательно каждую последнею 1 на 0. (Последние единицы это идущие подряд 1, начиная с конца до первого нуля или до начала вектора, если нулей нет). Если в векторе последний 0, то новые решения не получаем, так как в этом случае будут повторяющиеся решения, полученные ранее. Полное дерево решений для вектора из 4-х элементов (рис. 1). Вершиной дерева является единичное решение, на основе этого

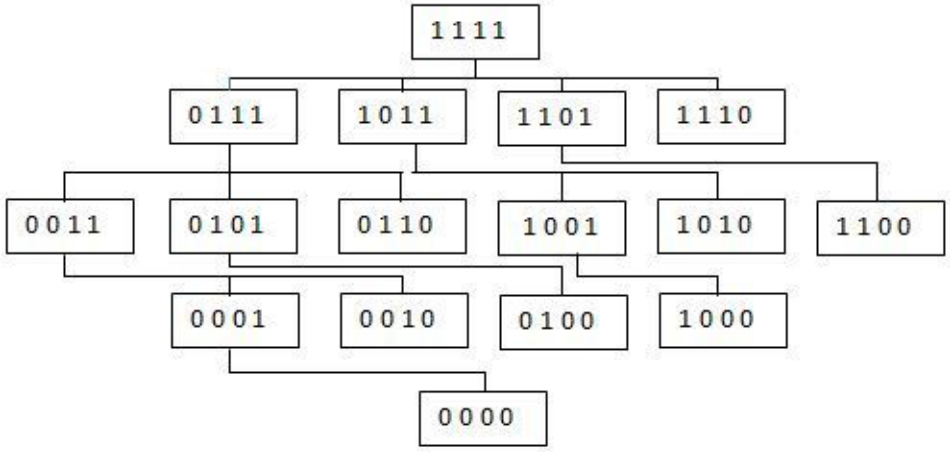

Рис. 1. Полное дерево решений для вектора из 4-х элементов

решения получаем на первом шаге четыре новых решения, представленных на 2-ом уровне дерева.

Полное дерево (рис. 1) соответствует алгоритму полного перебора, В таком виде дерево не строится, оно строится в сокращенном виде.

Для каждого нового решения проверяем его допустимость, если оно допустимое, то путь на дереве завершается, так как в соответствии с аксиомой 2, это не приведет к улучшению значения показателя качества. Если значение показателя (1) для допустимого решения больше, чем рекорд, то полученное решение запоминаем как рекорд. Также прерываем путь на новом недопустимом решении, если значение показателя качества для него не лучше, чем текущий рекорд, так при замене 1 на 0 значение показателя не улучшится.

Обработку каждого решения удобнее выполнять с помощью рекурсивной функции, если полученное новое решение не является концом пути (листом дерева), то для него вызываем эту же функцию.

Таким образом, некоторые пути полного дерева завершаются досрочно. В результате работы алгоритма оптимальное решение будет храниться в текущем рекордном решении.

Алгоритм частичного перебора, начиная с нулевого решения

В этом алгоритме на вершине дерева находится решение, состоящее из всех 0, это решение является допустимым по ограничениям. Первоначально считает рекордное значение показателя качества 0, рекордное решение пустое. На следующем уровне дерева, получаем новые решения, заменив один из последних нулей на единицу, аналогично как предыдущем алгоритме меняли 1 на 0. Полное дерево решений для вектора из 4-х элементов (рис. 2).

При движении по дереву путь прерываем досрочно, если все новые решения являются недопустимыми по ограничениям (аксиома 1). В этом случае для рассматриваемого допустимого решения вычисляем значение показателя качества, если значение больше, чем рекорд, то полученное решение сохраняем как рекордное, а значение показателя качества как рекорд. В результате работы алгоритма при завершении просмотра неполного дерева оптимальное решение будет храниться в текущем рекордном решении.

Представленные алгоритмы, частично аналогичны алгоритму Балаша [20], в алгоритме Балаша путь на

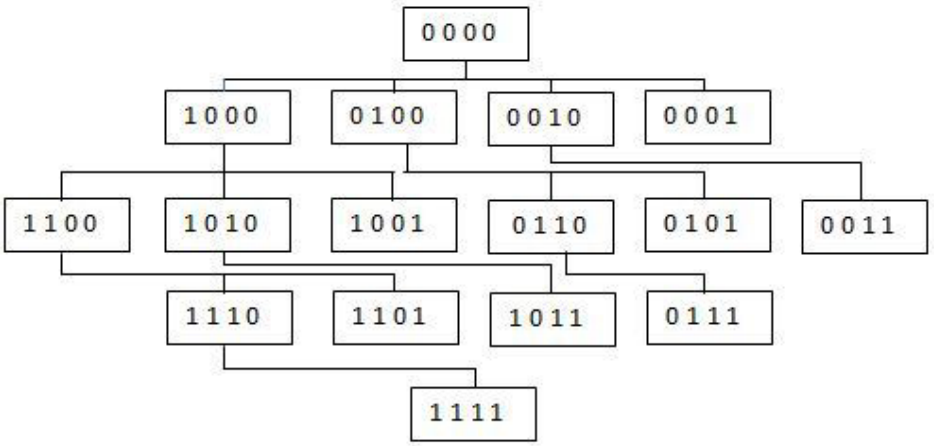

Рис. 2. Полное дерево решений для вектора из 4-х элементов

Таблица 1

Данные по программным средствам защиты

№ п/п, индекс k, средства $k \in \Lambda$	Название программного средства	Ресурсы для процессов (одинаковые значения для всех объектов)			Вероятности защиты от угроз, $p_{jk}, \forall j \in M,\ k \in \Lambda$		
		Стоимость, $v_{*1k}^{(о)}$, руб.	Загрузка процессора, $v_{*1k}^{(ч)}$, %	Память, $v_{*2k}^{(ч)}$, %	Названия угроз		
					Целостность	Доступность	Конфиденциальность
1	Антивирусное средство 1	2000	5	5	0.70	0.80	0.50
2	Антивирусное средство 2	2500	7	5	0.75	0.90	0.50
3	Антивирусное средство 3	3000	10	7	0.80	0.90	0.60
4	Криптографическое средство 1	2500	10	7	0.00	0.00	0.95
5	Криптографическое средство 2	2300	5	5	0.00	0.00	0.90
6	Программа для контроля целостности данных 1	1700	10	5	0.95	0.10	0.00
7	Программа для контроля целостности данных 2	1500	5	2	0.9	0.1	0

подобном дереве завершался досрочно по двум причинам:

- получаемые новые решения заменой 0 на 1 из некоторого рассматриваемого решения не удовлетворяют ограничениям;
- целевая функция при переходе к следующим решениям не улучшится.

Аналогичные идеи используются в предложенных алгоритмах с учетом того, что целевая функция нелинейная и разные начальные решения на вершине дерева.

3.2. Приближенные алгоритмы

Предложенные выше точные алгоритмы, несмотря на то, что они лучше полного перебора, имеют экспоненциальную вычислительную сложность. Рассмотрим два алгоритма с полиномиальной трудоемкостью, основанные на принципе «жадного» алгоритма.

В первом алгоритме начнем просмотр решений, начиная с единичного решения, которое является недопустимым. На каждом шаге алгоритма одну 1 заменяем на 0, при этом выбираем из всех 1 такую замену, которая дает максимум показателю (1). Если на каком то шаге получено допустимое решение, то работу алгоритма завершаем, полученное решение есть результат работы алгоритма.

Во втором алгоритме начнем просмотр решений, начиная с нулевого решения, которое является допустимым. На каждом шаге алгоритма один 0 заменяем на 1, при этом выбираем из всех 0 такую замену, которая дает максимум показателю (1) и решение является допустимым по ограничениям. Если на каком то шаге получено допустимое решение, в котором при замене любого 0 на 1, получаем недопустимое решение, то работу алгоритма завершаем, полученное решение есть результат работы алгоритма.

4. Пример решения задачи и эксперименты

4.1. Пример решения задачи

Для демонстрации рассмотрим пример решения задачи небольшой размерности на условно-реальных исходных данных. Названия приложений и их производители не указываются для исключения рекламы и антирекламы, данные по ущербу и вероятностям проведения атак приведены для некоторой условной организации.

Приведены данные по программным средствам, которые могут быть установлены на серверах организации (табл. 1). Данные представлены по 7 программным средствам: по трем антивирусам, входящим в группу 1, по двум криптографическим средствам, входящим в группу 2, и по двум средствам контроля целостности, входящим в группу 3. Первоначально идут столбцы с ресурсами, требуемыми для работы средств, приведен один общий ресурс – стоимость, измеряется в рублях, и два частных ресурса – загрузка процессора и использование памяти, измеряются в процентах от всего ресурса сервера. Для сокращения исходных данных,

Таблица 2

Данные по серверам организации

№ п/п, индекс -го сервера, $i \in N$	Название	Оценка ущерба, $w_{ij}, \forall\, i \in N,\ j \in M$			Вероятность (или возможность) реализации j-ой атаки на i-ый сервер, $p_{ij}^{(a)}, \forall\ i \in N,\ j \in M$			Максимальное значение j-го частного ресурса на i-ом сервере, $V_{ij}^{(ч)}, \forall\, i \in N,\ j \in L^{(ч)}$	
		Номера угроз			Номера угроз			Загрузка	Память
		1	2	3	1	2	3		
1	Сервер управления кадров для хранения персональных данных	10000	2000	20000	0.5	0.5	0.90	30	20
2	Сервер для обеспечения работы сайта организации с общедоступными данными в сети Интернет	10000	10000	0	0.7	0.8	0.00	20	20
3	Сервер для хранения конфиденциальных данных организации	15000	15000	30000	0.8	0.8	0.99	30	20

Таблица 3

Результаты решения задачи точным методом частичного перебора, начиная с единичного решения

№ приложения / № сервера	1	2	3	4	5	6	7
1	0	0	1	1	0	1	0
2	0	0	1	0	0	1	0
3	0	0	1	1	0	1	0

Таблица 4

Результаты решения задачи точным методом частичного перебора, начиная с нулевого решения

№ приложения / № сервера	1	2	3	4	5	6	7
1	0	1	0	1	0	1	0
2	0	1	0	0	0	1	0
3	0	1	0	1	0	1	0

считаем, что эти ресурсы для всех серверов одинаковы (звездочка вместо индекса сервера). Далее представлены вероятности предотвращения реализаций угроз безопасности. Представлены три глобальные угрозы: целостность, доступность, конфиденциальность. В реальности эти угрозы могут разбиваться на более частные угрозы.

Приведены данные по серверам организации (табл. 2). Рассмотрено три сервера: сервер управления кадров для хранения персональных данных, сервер для обеспечения работы сайта организации с общедоступными данными в сети Интернет, сервер для хранения конфиденциальных данных организации. Вначале в столбцах представлены оценки ущерба для серверов при реализации угроз (угрозы идут под номерами: 1- целостность, 2- доступность, 3- конфиденциальность), затем вероятности (возможности) реализации этих угроз на сервера, последние два столбца значение ресурсов процессорного времени и памяти на серверах в процентах (что осталось от целевых приложений, которые обязательно запущены на серверах). Значение общего ресурса максимальной стоимости закупаемого программного обеспечения – 20000 рублей.

В результате решения задачи найденное оптимальное значение показателя качества равно 87015 руб. Результаты решения задачи точным методом частичного перебора, начиная с единичного решения (табл. 3).

Результаты решения задачи точным методом частичного перебора, начиная с нулевого решения (табл. 4).

При решении задачи приближенными методами, было получено точное решение, такое как в (табл. 3).

Следует отметить, что получение двух разных оптимальных решений с одним значением показателя качества объяснимо тем, что в исходных данных часто встречаются одинаковые значения параметров.

4.2. Эксперименты на исходных данных, полученных с помощью генераторов псевдослучайных чисел

Рассмотрим результаты тестирования алгоритмов на исходных данных, сгенерированных генераторами псевдослучайных чисел (ГПСЧ).

Эксперименты проводились на ноутбуке с процессором Intel(R) Core(TM) i3-6006U CPU @ 2.00GHz, 2000 МГц, ядер: 2, логических процессоров: 4, оперативной памяти 4 ГБ, операционная система Windows 10, среда разработки Microsoft Visual Studio 2019, язык программирования Си++.

Эксперименты по сравнения точных методов с точки зрения вычислительной сложности алгоритмов

Предложенные выше точные методы имеют экспоненциальную вычислительную сложность, можно предположить, что если ресурсы не позволяют в искомом векторе получить много единиц, то более эффективным с точки зрения вычислительной сложности при таких исходных данных будет перебор, начиная с нулевого решения. Если в векторе X будет много единиц, то более эффективным будет перебор, начиная с единичного решения.

Введем понятие коэффициента обеспеченности ресурсами по аналогии с [21]. Коэффициент обеспеченности одного ресурса содержательно определим как отношение объема ресурса, который есть в наличии к объему этого ресурса, который требуется, чтобы можно были использовать все приложения на всех объектах, т.е. состоит из всех единиц.

Рассмотрим, как влияет этот коэффициент для различных типов ресурсов. Ниже представлены результаты экспериментов по определению зависимости времени решения задач разными точными алгоритмами в зависимости от коэффициента обеспеченности ресурсами (рис. 3). Коэффициент подбирался одинаковый для всех ресурсов, как для частных, так и для общих. Исходные данные генерировались с помощью ГПСЧ так, чтобы обеспечить заданное значение коэффициента. Эксперименты проводились при следующих исходных данных: число объектов – 5, число угроз – 5, число приложений – 5, число общих ресурсов – 1, число частных ресурсов – 2.

Далее представлены графики зависимости времени решения задач алгоритмами от коэффициента обеспеченности общим ресурсом при различных коэффициентах обеспеченности частными ресурсами: 0.2, 0.5, 0.8 (рис.4, рис. 5, рис. 6).

Представлены графики зависимости времени решения задач алгоритмами от коэффициента обеспеченности частными ресурсами при различных коэффициентах обеспеченности общим ресурсом: 0.2, 0.5, 0.8 (рис.7, рис. 8, рис. 9).

Из представленных графиков можно сделать вывод, что для разных типов ресурсов важно минимальное значение коэффициента обеспеченности. Для нескольких ресурсов будем использовать минимальное значение этого коэффициента среди этих ресурсов (частных и общих):

$$K_r = \min\left\{ \min_{j \in L^{(ч)}} \left\{ \frac{\sum_{i \in N} V_{ij}^{(ч)}}{\sum_{i \in N} \sum_{k \in \Lambda} v_{ijk}^{(ч)}} \right\}, \min_{j \in L^{(о)}} \left\{ \frac{V_j^{(о)}}{\sum_{i \in N} \sum_{k \in \Lambda} v_{ijk}^{(о)}} \right\} \right\}. \quad (5)$$

Значение коэффициента, вычисляемое по формуле (5) определяет число единиц в решениях, которые являются допустимыми по ресурсам. Чем меньше коэффициент обеспеченности ресурсов, тем меньше единиц в допустимых решениях.

Можно примерно считать, что точка, когда два алгоритма обеспечивают примерно одинаковое время решения, находится для коэффициента обеспеченности ресурсов в интервале от 0.4 до 0.6. Если значение коэффициента меньше, чем левая граница интервала, то предпочтительнее использовать метод частичного перебора, начиная с нулевого решения. Если значение коэффициента больше, чем правая граница интервала, то предпочтительнее использовать метод частичного перебора, начиная с единичного решения.

DOI:10.21681/2311-3456-2020-05-30-44

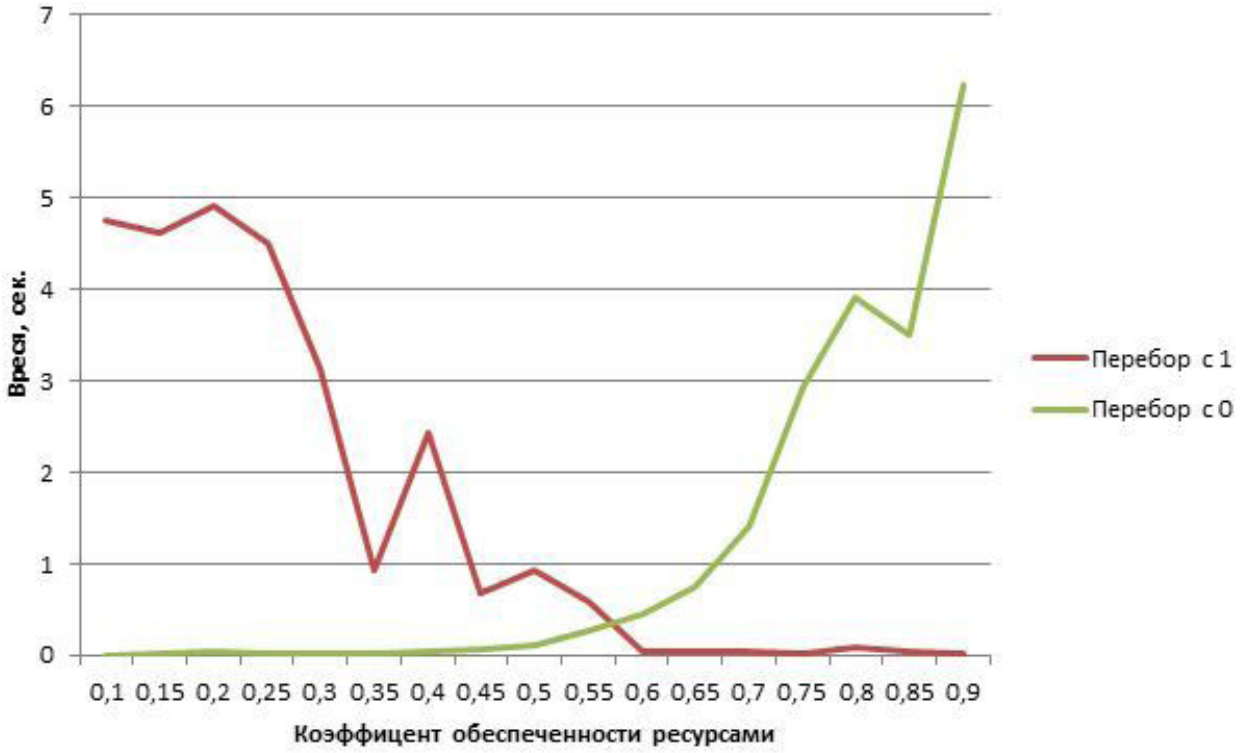

Рис. 3. Зависимость времени решения от коэффициента обеспеченности ресурсами

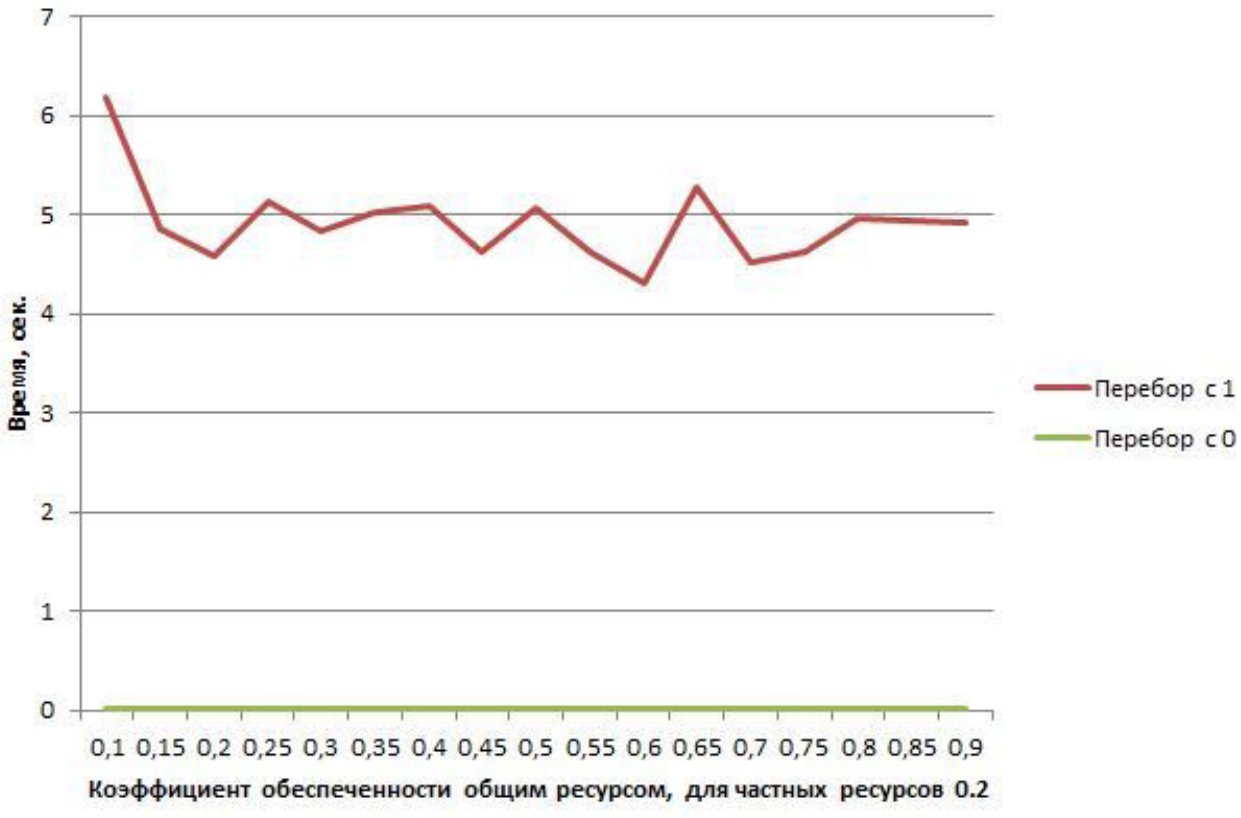

Рис.4 Зависимость времени решения от коэффициента обеспеченности общим ресурсом (коэффициент обеспеченности частными ресурсами 0.2)

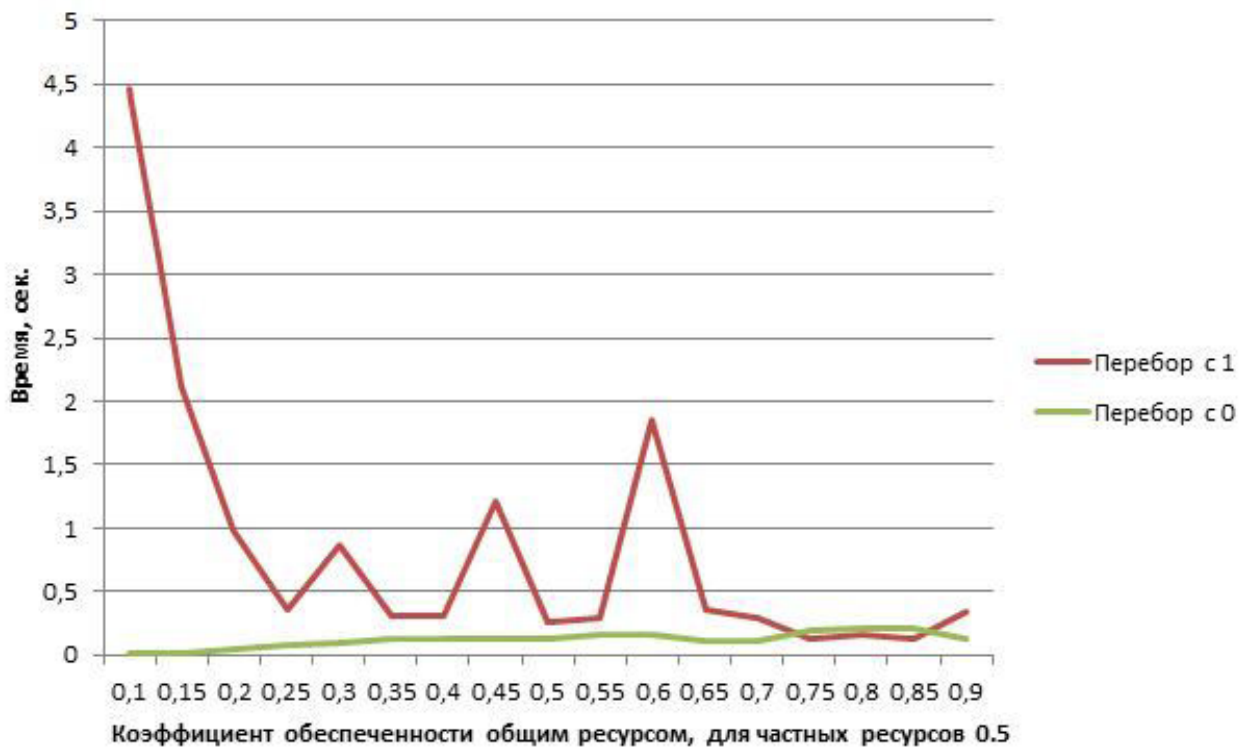

Рис.5 Зависимость времени решения от коэффициента обеспеченности общим ресурсом (коэффициент обеспеченности частными ресурсами 0.5)

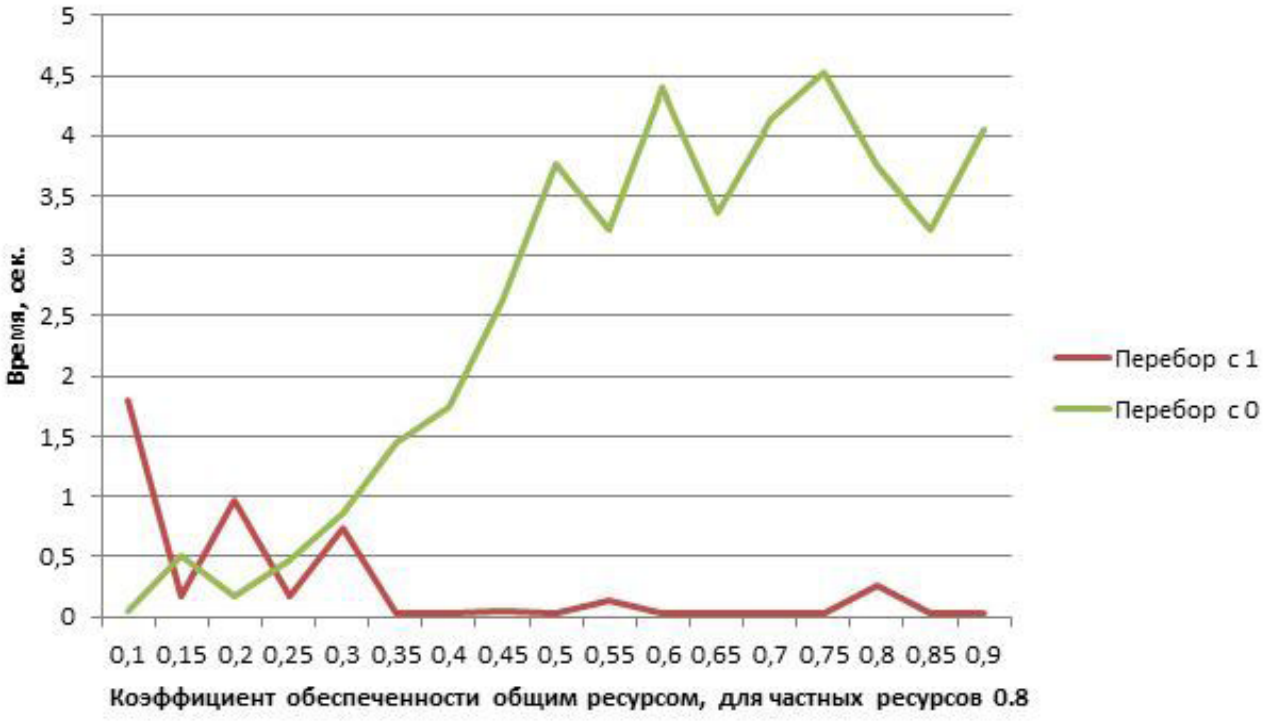

Рис.6 Зависимость времени решения от коэффициента обеспеченности общим ресурсом (коэффициент обеспеченности частными ресурсами 0.8)

DOI:10.21681/2311-3456-2020-05-30-44

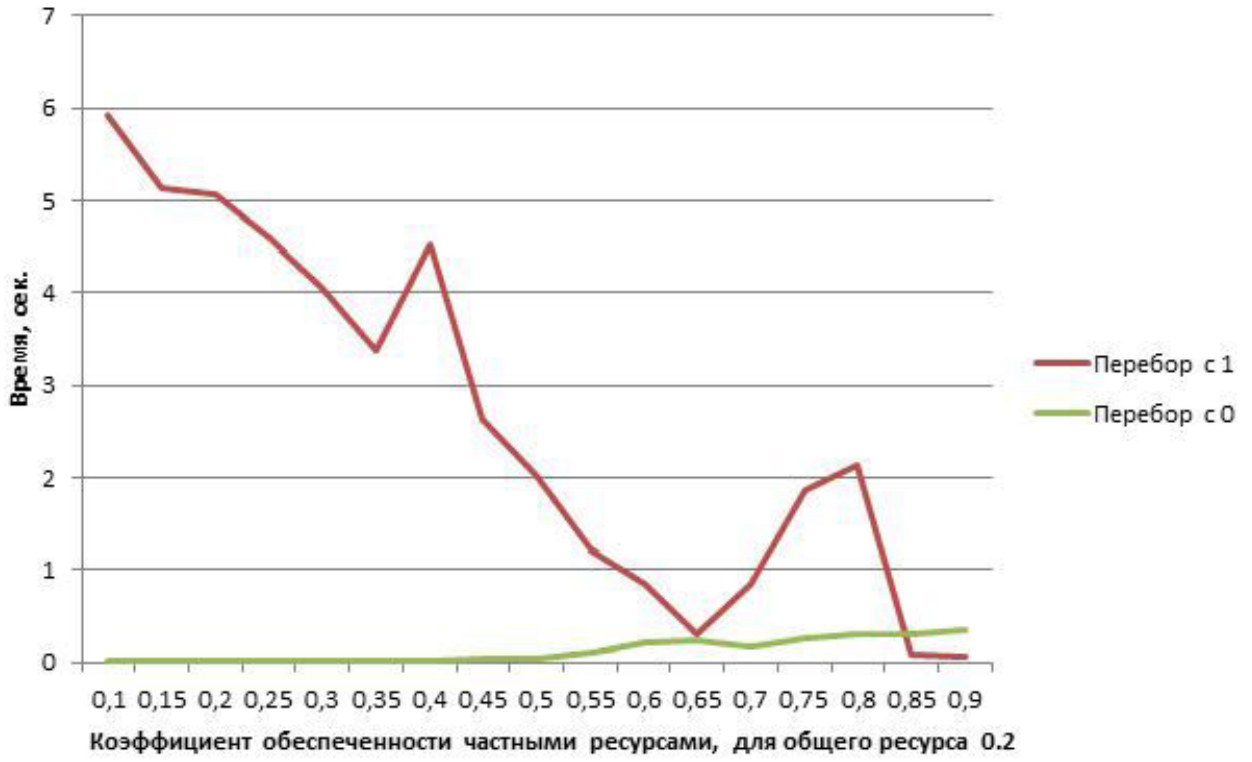

Рис.7 Зависимость времени решения от коэффициента обеспеченности частными ресурсами (коэффициент обеспеченности частным ресурсом 0.2)

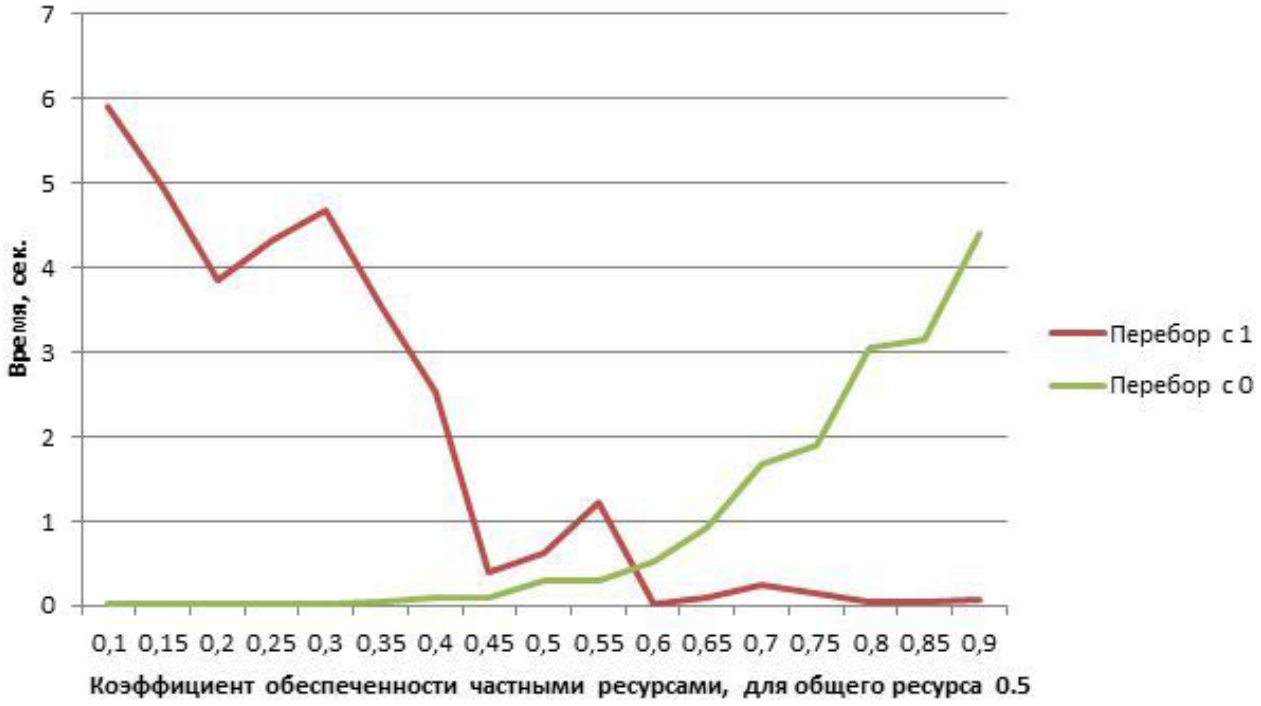

Рис.8 Зависимость времени решения от коэффициента обеспеченности частными ресурсами (коэффициент обеспеченности частным ресурсом 0.5)

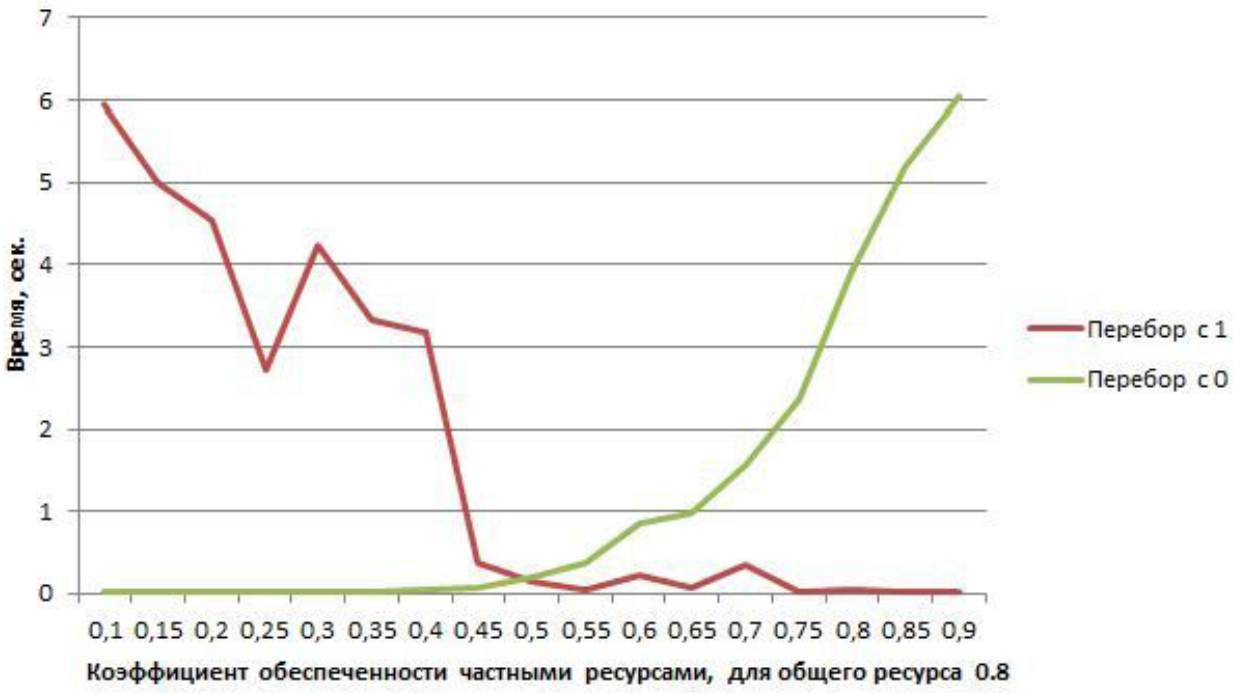

Рис.9 Зависимость времени решения от коэффициента обеспеченности частными ресурсами (коэффициент обеспеченности частным ресурсом 0.8)

Эксперименты по оценке точности приближенных методов

В результате экспериментов была проведена серия из 100 испытаний при следующих исходных данных: число объектов – 6, число угроз – 5, число приложений – 5, число общих ресурсов – 1, число частных ресурсов – 2. Получены следующие оценки средней относительной погрешности значения показателя для полученного решения по сравнению со значением показателя для решения, полученного точным методом. Жадный алгоритм поиска решений, начиная с единичного решения: 4.24 %. Жадный алгоритм поиска решений, начиная с нулевого решения: 0.87 %.

Выводы

Сформулирована и исследована задача выбора приложений (процессов) для защиты информации на объектах (серверах) вычислительной системы при ограничениях на ресурсы. Использован подход, что выбираемые приложения (процессы) являются служебными или необязательными и они используют ресурсы, оставшиеся от основных (обязательных) процессов. Предложено рассматривать общие ресурсы для всех серверов, например, финансовый ресурс и частные ресурсы для каждого отдельного сервера, например, вычислительные ресурсы. Математическая постановка задачи являлась задачей булевого программирования с нелинейным показателем качества и линейными ограничениями. Предложены точные алгоритмы решения задачи на основе идей метода Балаша: алгоритм, основанный на частичном переборе, начиная с нулевого решения, и алгоритм, основанный на частичном переборе, начиная с единичного решения. Предложены приближенные алгоритмы решения задачи с полиномиальной вычислительной сложностью на основе «жадного алгоритма»: алгоритм поиска, начиная с нулевого решения, и алгоритм поиска, начиная с единичного решения.

Представлен пример решения задачи небольшой размерности для условно-реальных исходных данных. Проведены эксперименты на исходных данных, полученных с помощью генераторов псевдослучайных чисел. Два точных алгоритма были экспериментально исследованы на предмет выбора одного из алгоритмов для снижения времени решения задачи в зависимости от исходных данных, для этого введено понятие – коэффициент обеспеченности ресурсов. Сформулированы следующие рекомендации: если значение коэффициента меньше 0.4, то предпочтительнее использовать метод частичного перебора, начиная с нулевого решения, значение коэффициента больше, чем 0.6, то предпочтительнее использовать метод частичного перебора, начиная с единичного решения, в интервале 0.4-0.6 два алгоритма могут показывать одинаковые результаты.

Проведены экспериментальные исследования по выявлению относительной погрешности двух приближенных алгоритмов: жадный алгоритм поиска решений, начиная с единичного решения, в среднем показал относительную погрешность 4.24 %, а жадный алгоритм поиска решений, начиная с нулевого решения, в среднем показал относительную погрешность 0.87 %.

Достоверность полученных результатов подтверждается корректностью математической постановки задачи, очевидной содержательной интерпретацией, как постановки задачи, так и получаемых решений, а также сравнением решений получаемых разными алгоритмами.

DOI:10.21681/2311-3456-2020-05-30-44

Литература

1. Зангиев Т.Т., Романенко А.В., Позднякова Е.Г. Выбор средств защиты информации при многих критериях с нечетким описанием // Свидетельство о регистрации программы для ЭВМ RU 2019614539, 05.04.2019.
2. Зангиев Т.Т., Туркин Е.А., Чернецова Т.В., Корх И.А. Оптимальный выбор средств криптографической защиты для банковских систем в нечеткой среде // Свидетельство о регистрации программы для ЭВМ RU 2019616070, 17.05.2019.
3. Зангиев Т.Т., Постельный Е.М. Оптимальный выбор средств защиты информации от несанкционированного доступа методом анализа иерархий // Химия, физика, биология, математика: теоретические и прикладные исследования: сборник статей по материалам XI-XII международной научно-практической конференции. 2018. С. 33-40.
4. Касенов А.А., Кустов Е.Ф., Магазев А.А., Цырульник В.Ф. Марковская модель оптимизации средств защиты информации // Динамика систем, механизмов и машин. 2019. Т. 7. № 4. С. 77-84.
5. Павликов С.Н., Убанкин Е.И., Коломеец В.Ю., Пленник М.Д. Разработка многопараметрической последовательно-параллельной матричной системы защиты информационной сети // Наукоемкие технологии в космических исследованиях Земли. 2019. Т. 11. № 5. С. 39-47. DOI: 10.24411/2409-5419-2018-10286
6. Кащенко А.Г. Модель выбора варианта системы защиты информации для распределенной вычислительной сети предприятия // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. 2010. № 2. С. 46-49.
7. Nilotpal Chakraborty, Ezhil Kalaimannan. Minimum cost security measurements for attack tree-based threat models in smart grid // 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON). DOI: 10.1109/UEMCON.2017.8249049
8. Zan Li, Shiqi Gong, Chengwen Xing, Zesong Fei, Xinge Yan. Multi-Objective Optimization for Distributed MIMO Networks // IEEE Transactions on Communications 2017. Vol. 65. Iss. 10. P. 4247-4259. DOI: 10.1109/TCOMM.2017.2722478
9. Peng Wei, Yufeng Li, Zhen Zhang, Tao Hu, Ziyong Li, Diyang Liu. An Optimization Method for Intrusion Detection Classification Model Based on Deep Belief Network // IEEE Access. 2019. Vol. 7. P. 87593-87605. DOI: 10.1109/ACCESS.2019.2925828
10. Ali Safaa Sadiq, Basem Alkazemi, Seyedali Mirjalili, Noraziah Ahmed, Suleman Khan, Ihsan Ali, Al-Sakib Khan Pathan, Kayhan Zrar Ghafoor. An Efficient IDS Using Hybrid Magnetic Swarm Optimization in WANETs // IEEE Access. 2018. Vol. 6. P. 29041-29053. DOI: 10.1109/ACCESS.2018.2835166
11. Hong Zhang, Yun Cao, Xianfeng Zhao. A Steganalytic Approach to Detect Motion Vector Modification Using Near-Perfect Estimation for Local Optimality // IEEE Transactions on Information Forensics and Security. 2017. Vol. 12. Iss. 2. P. 465-478. DOI: 10.1109/TIFS.2016.2623587
12. Tuan Anh Le, Quoc-Tuan Vien, Huan X. Nguyen, Derrick Wing Kwan Ng, Robert Schober. Robust Chance-Constrained Optimization for Power-Efficient and Secure SWIPT Systems // IEEE Transactions on Green Communications and Networking. 2017. Vol. 1. Iss. 3. P. 333-346. DOI: 10.1109/TGCN.2017.2706063
13. Xiaobo Zhou, Qingqing Wu, Shihao Yan, Feng Shu, Jun Li. UAV-Enabled Secure Communications: Joint Trajectory and Transmit Power Optimization // IEEE Transactions on Vehicular Technology. 2019. Vol. 68. Iss. 4. P. 4069-4073. DOI: 10.1109/TVT.2019.2900157
14. Jiaxin Yang, Qiang Li, Yunlong Cai, Yulong Zou, Lajos Hanzo, Benoit Champagne. Joint Secure AF Relaying and Artificial Noise Optimization: A Penalized Difference-of-Convex Programming Framework // IEEE Access. 2016. Vol. 4. P. 10076-10095. DOI: 10.1109/ACCESS.2016.2628808
15. Bykov A.Yu., Grishunin M.V., Krygin I.A. Saddle point search algorithm for the problem of site protection level assignment based on search of simplices // Вестник Московского государственного технического университета им. Н.Э. Баумана. Серия: Приборостроение. 2019. № 2 (125). С. 22-39. DOI: 10.18698/0236-3933-2019-2-22-39
16. Быков А.Ю., Крыгин И.А., Гришунин М.В. Алгоритм поиска седловой точки в смешанных стратегиях на основе модификации метода Брауна-Робинсона для решения задачи выбора защищаемых объектов // Безопасные информационные технологии Сборник трудов Девятой всероссийской научно-технической конференции. 2018. С. 33-38.
17. Быков А.Ю., Гришунин М.В., Крыгин И.А. Игровая задача выбора защищаемых объектов и исследование алгоритма поиска седловой точки на основе модификации метода Брауна-Робинсона // Вопросы кибербезопасности. 2019. № 2 (30). С. 2-12. DOI: 10.21681/2311-3456-2019-2-2-12
18. Хачатрян М.Г., Ключарев П.Г. Распознавание ботов в онлайновых социальных сетях при помощи алгоритма «Случайный лес» // Машиностроение и компьютерные технологии. 2019. № 4. С. 24-41. DOI: 10.24108/0419.0001473
19. Ключарёв П.Г. Детерминированные методы построения графов Рамануджана, предназначенных для применения в криптографических алгоритмах, основанных на обобщённых клеточных автоматах // Прикладная дискретная математика. 2018. № 42. С. 76-93. DOI 10.17223/20710410/42/6
20. Басараб М.А., Вельц С.В. Методы оптимизации и исследование операций в области информационной безопасности: Методические указания к выполнению лабораторных работ по дисциплине «Методы оптимизации и исследования операций». М.: МГТУ им. Н.Э. Баумана, 2015. 64 с. Режим доступа: http://ebooks.bmstu.press/catalog/117/book967.html
21. Кошман А.А. Алгоритмы поиска допустимых решений для построения матрицы игры в задаче выбора объектов защиты при ограничениях на ресурсы // Политехнический молодежный журнал. 2019. № 4 (33). С. 7-17. DOI: 10.18698/2541-8009-2019-4-471

THE PROBLEM OF SELECTING COMPUTING PROCESSES THAT PROVIDE INFORMATION PROTECTION FOR SERVERS OF A DISTRIBUTED SYSTEM AND ALGORITHMS FOR ITS SOLUTION

Bykov A.Yu.[4], Krygin I.A.[5], Grishunin M.V.[6]

Purpose: *providing information security on servers of various destinations of an automated system based on the formulation of an optimization task setting for selecting auxiliary processes for information protection, developing and researching algorithms for solving this problem.*

Method: *To solve the problem, the authors propose two exact algorithms for incomplete search with exponential computational complexity, based on the ideas of the Balash method. One algorithm starts with a solution consisting of all ones, and the second algorithm starts with a solution consisting of all zeros. Also proposed are two approximate algorithms with polynomial complexity, based on the ideas of the "greedy" algorithm, one algorithm starts the search from a zero solution, the other from all-one solution.*

Result: *Mathematical model and algorithms for solving the problem of selecting processes to protect information in conditions of limited computing resources of servers. The process selection model is a Boolean programming problem with a non-linear quality indicator and linear constraints. The indicator provides an estimate of the damage prevented when using the selected processes, taking into account the probability or possibility of various attacks on servers, the value of stored data, and the probability of protection from attacks using processes. During the experiments, recommendations were developed for choosing one of two algorithms (search starting from the zero solution and search starting from all-one solution) among pairs of exact and approximate algorithms depending on the availability of resources in order to reduce the time to solve the problem.*

Keywords: *information security, discrete optimization, boolean programming, resource ratio, computational complexity of the algorithm, approximate solution.*

References

1. Zangiev T.T., Romanenko A.V., Pozdnyakova E.G. Vybor sredstv zashchity informacii pri mnogih kriteriyah s nechetkim opisaniem. Svidetel'stvo o registracii programmy dlya EVM RU 2019614539, 05.04.2019. (in Russ.)
2. Zangiev T.T., Turkin E.A., CHernecova T.V., Korh I.A. Optimal'nyj vybor sredstv kriptograficheskoj zashchity dlya bankovskih sistem v nechetkoj srede. Svidetel'stvo o registracii programmy dlya EVM RU 2019616070, 17.05.2019. (in Russ.)
3. Zangiev T.T., Postel'nyj E.M. Optimal'nyj vybor sredstv zashchity in-formacii ot nesankcionirovannogo dostupa metodom analiza ierarhij // Himiya, fizika, biologiya, matematika: teoreticheskie i prikladnye issledovaniya: cbornik statej po materialam XI-XII mezhdunarodnoj nauchno-prakticheskoj konfe-rencii, 2018, pp. 33-40. (in Russ.)
4. Kasenov A.A., Kustov E.F., Magazev A.A., Cyrul'nik V.F. Markovskaya model' optimizacii sredstv zashchity informacii [Markov model for optimization of information security remedies] // Dinamika sistem, mekhanizmov i mashin, 2019, vol. 7, no. 4, pp. 77-84. (in Russ.)
5. Pavlikov S.N., Ubankin E.I., Kolomeec V.YU., Plennik M.D. Razrabotka mnogoparametricheskoj posledovatel'no-parallel'noj matrichnoj sistemy zashchity informacionnoj seti // Naukoemkie tekhnologii v kosmicheskih issledovaniyah Zemli, 2019, vol. 11, no. 5, pp. 39-47. (in Russ.) DOI: 10.24411/2409-5419-2018-10286
6. Kashchenko A.G. Model' vybora varianta sistemy zashchity informacii dlya raspredelennoj vychislitel'noj seti predpriyatiya // Vestnik Voronezhskogo gosudarstvennogo universiteta. Seriya: Sistemnyj analiz i informacionnye tekhnologii, 2010, no 2, pp. 46-49. (in Russ.)
7. Nilotpal Chakraborty, Ezhil Kalaimannan. Minimum cost security measurements for attack tree based threat models in smart grid // 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON). DOI: 10.1109/UEMCON.2017.8249049
8. Zan Li, Shiqi Gong, Chengwen Xing, Zesong Fei, Xinge Yan. Multi-Objective Optimization for Distributed MIMO Networks // IEEE Transactions on Communications 2017, vol. 65, iss. 10, pp. 4247-4259. DOI: 10.1109/TCOMM.2017.2722478
9. Peng Wei, Yufeng Li, Zhen Zhang, Tao Hu, Ziyong Li, Diyang Liu. An Optimization Method for Intrusion Detection Classification Model Based on Deep Belief Network // IEEE Access, 2019, vol. 7, pp. 87593-87605. DOI: 10.1109/ACCESS.2019.2925828
10. Ali Safaa Sadiq, Basem Alkazemi, Seyedali Mirjalili, Noraziah Ahmed, Suleman Khan, Ihsan Ali, Al-Sakib Khan Pathan, Kayhan Zrar

4 Alexander Bykov, Ph.D., associate Professor, Department of Information Security, BMSTU, Moscow, Russia. Email: abykov@bmstu.ru

5 Ivan Krygin, post-graduate student, Department of Information Security, BMSTU, Moscow, Russia. Email: krygin.ia@gmail.com

6 Maxim Grishunin, post-graduate student, Department of Information Security, BMSTU, Moscow, Russia. Email: grishunin-mv@ya.ru

Ghafoor. An Efficient IDS Using Hybrid Magnetic Swarm Optimization in WANETs // IEEE Access, 2018, vol. 6, pp. 29041-29053. DOI: 10.1109/ACCESS.2018.2835166

11. Hong Zhang, Yun Cao, Xianfeng Zhao. A Steganalytic Approach to Detect Motion Vector Modification Using Near-Perfect Estimation for Local Optimality // IEEE Transactions on Information Forensics and Security, 2017, vol. 12, iss. 2, pp. 465-478. DOI: 10.1109/TIFS.2016.2623587
12. Tuan Anh Le, Quoc-Tuan Vien, Huan X. Nguyen, Derrick Wing Kwan Ng, Robert Schober. Robust Chance-Constrained Optimization for Power-Efficient and Secure SWIPT Systems // IEEE Transactions on Green Communications and Networking, 2017, vol. 1, iss. 3, pp. 333-346. DOI: 10.1109/TGCN.2017.2706063
13. Xiaobo Zhou, Qingqing Wu, Shihao Yan, Feng Shu, Jun Li. UAV-Enabled Secure Communications: Joint Trajectory and Transmit Power Optimization // IEEE Transactions on Vehicular Technology, 2019, vol. 68, iss. 4, pp. 4069-4073. DOI: 10.1109/TVT.2019.2900157
14. Jiaxin Yang, Qiang Li, Yunlong Cai, Yulong Zou, Lajos Hanzo, Benoit Champagne. Joint Secure AF Relaying and Artificial Noise Optimization: A Penalized Difference-of-Convex Programming Framework // IEEE Access, 2016, vol. 4, pp. 10076-10095. DOI: 10.1109/ACCESS.2016.2628808
15. Bykov A.Yu., Grishunin M.V., Krygin I.A. Saddle point search algorithm for the problem of site protection level assignment based on search of simplices // Vestnik Moskovskogo gosudarstvennogo tekhnicheskogo universiteta im. N.E. Baumana. Ser. Priborostroenie [Herald of the Bauman MSTU. Ser. Instrument Engineering], 2019, no. 2 (125), pp. 22-39. DOI: 10.18698/0236-3933-2019-2-22-39
16. Bykov A.Yu., Krygin I.A., Grishunin M.V. Algoritm poiska sedlovoj tochki v smeshannyh strategiyah na osnove modifikacii metoda Brauna-Robinsona dlya resheniya zadachi vybora zashchishchaemyh ob"ektov [The algorithm of saddle point search in mixed strategies based *on Brown–Robinson method modification to solve problem of assets to protect selection*] // V sbornike: Bezopasnye informacionnye tehno-logii Sbornik trudov Devjatoj vserossijskoj nauchno-tehnicheskoj konferencii, 2018, pp. 33-38.
17. Bykov A.Yu., Grishunin M.V., Krygin I.A. Igrovaya zadacha vybora zashchishchaemyh ob"ektov i issledovanie algoritma poiska sedlovoj tochki na osnove modifikacii metoda Brauna-Robinsona [*The game problem of selection of assets to protect and research of saddle point search algorithm based on Brown-Robinson method modification*] // Voprosy kiberbezopasnosti [*Cybersecurity issues*], 2019, no. 2 (30). pp. 2-12. DOI: 10.21681/2311-3456-2019-2-2-12
18. Hachatryan M.G., Klyucharev P.G. Raspoznavanie botov v onlajnovyh social'-nyh setyah pri pomoshchi algoritma "Sluchajnyj les" [*Bots recognition in social networks using the Random Forest algorithm*] // Mashinostroenie i komp'yuternye tekhnologii, 2019, no. 4, pp. 24-41. DOI: 10.24108/0419.0001473
19. Klyucharyov P.G. Determinirovannye metody postroeniya grafov Ramanudzha-na, prednaznachennyh dlya primeneniya v kriptograficheskih algoritmah, osnovan-nyh na obobshchyonnyh kletochnyh avtomatah [*Deterministic methods of Ramanujan graph construction for use in cryptographic algorithms based on generalized cellular automata*] // Prikladnaya diskretnaya matematika, 2018, no. 42, pp. 76-93. DOI 10.17223/20710410/42/6
20. Basarab M.A., Vel'c S.V. Metody optimizacii i issledovanie operacij v oblasti informacionnoj bezopasnosti: Metodicheskie ukazanija k vypolneniju laboratornyh rabot po discipline «Metody optimizacii i issledovanija operacij». M.: MGTU im. N.Je. Baumana, 2015, 64 p. Rezhim dostupa: http://ebooks.bmstu.press/catalog/117/book967.html
21. Koshman A.A. Algoritmy poiska dopustimyh reshenij dlya postroeniya mat-ricy igry v zadache vybora ob"ektov zashchity pri ogranicheniyah na resursy [*Algorithms of searching for admissible solutions to construct the matrix of the game in the problem of choice of protection objects under the restrictions on resources*] // Politekhnicheskij molodezhnyj zhurnal, 2019, no. 4 (33), pp. 7-17. DOI: 10.18698/2541-8009-2019-4-471